15. isnull() & notnull()

These functions are use to detect missing values in a DataFrame or a Series.
Essential for data cleaning and preprocessing as they help identify and handle missing or null values in your dataset.

isnull()

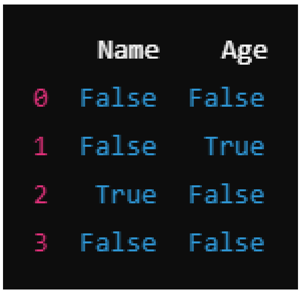
Detects missing values in a DataFrame or Series and returns a DataFrame or Series of the same shape, where each element is a Boolean indicating whether the corresponding element is missing(True) or not(False).

Ex:

```
import pandas as pd

# Example DataFrame
data = {
        'Name': ['Alice', 'Bob', None, 'David'],
    'Age': [24, None, 22, 32]
}
df = pd.DataFrame(data)

# Detect missing values
missing_values = df.isnull()
print(missing_values)
```



```
    Name    Age
0  False  False
1  False   True
2   True  False
3  False  False
```

---

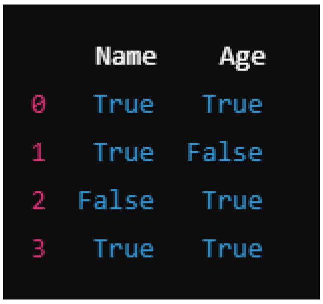
notnull()

Inverse of the `isnull()` function. It detects non-missing values in a DataFrame or Series and returns a DataFrame or Series of the same shape, where each element is a Boolean indicating whether the corresponding element is not missing (True) or missing (False).
Ex:

```
import pandas as pd

# Example DataFrame
data = {
        'Name': ['Alice', 'Bob', None, 'David'],
    'Age': [24, None, 22, 32]
}
df = pd.DataFrame(data)

# Detect missing values
missing_values = df.notnull()
print(missing_values)
```

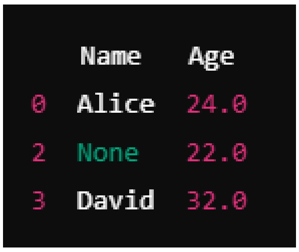


```
    Name    Age
0   True   True
1   True  False
2  False   True
3   True   True
```

Usage :-

1. Filtering data.
   Use these functions to filter out missing or non-missing values.

```
# Filter rows where 'Age' is not missing
df_age_notnull = df[df['Age'].notnull()]
print(df_age_notnull)
```

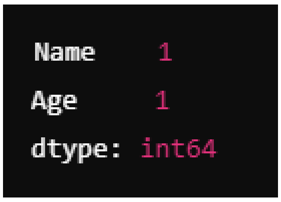


```
    Name   Age
0  Alice  24.0
2   None  22.0
3  David  32.0
```

- Boolean indexing is happening inside. We have filtered out rows which has missing values in the 'Age' column.

2. Counting missing values.
   Count the number of missing values in each column.

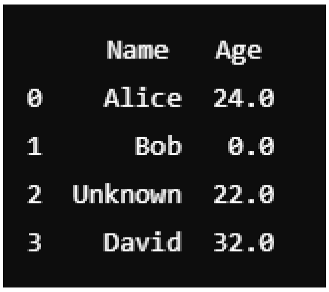
```
# Count missing values in each column
missing_count = df.isnull().sum()
print(missing_count)
```



```
Name     1
Age      1
dtype: int64
```

3. Handling missing values
   Able to replace missing values with a specific value or perform other operations to handle them.

```
df_filled = df.fillna({'Name': 'Unknown', 'Age': 0}) print(df_filled)
```



```
      Name   Age
0    Alice  24.0
1      Bob   0.0
2  Unknown  22.0
3    David  32.0
```